

Sensitivity and precision using the DM1 dataset

Alessia Buratin

October 4, 2021

1 Load the benchmarking results

We load the benchmarking results, which were produced by the script `glm_glmm_paired.R`.

```
basedir = "/blackhole/alessia/CircModel/"
# load(file = "/blackhole/alessia/CircModel/power/IPF_sensitivityPrecision_CCP2_glmglmm_30rep.RData")
load(file = "/blackhole/alessia/CircModel/robustness/ALZ_sensitivityPrecision_CCP2_glmglmm_30rep.RData.bk")
```

The evaluation set results are contained in the `resTes` object and the verification set results are contained in the `resHeldout` object, each a list, one element for each random replicate, of data frames which contain a column for each algorithm giving the adjusted p -values for each circRNA. For p -value adjustment, the `p.adjust` function was used with `method="BH"` (Benjamini-Hochberg correction), over only those genes with non-zero row sum.

```
library("ggplot2")
library("reshape")
library("data.table")

##
## Attaching package: 'data.table'
## The following object is masked from 'package:reshape':
##
##      melt

library("dplyr")

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:data.table':
##
##      between, first, last
## The following object is masked from 'package:reshape':
##
##      rename
## The following objects are masked from 'package:stats':
##
##      filter, lag
## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union

alpha=.1
getCalls <- function(m, alpha=NULL) {
  t(sapply(1:nreps, function(i) sapply(namesAlgos, function(algo) {
    sum(m[[i]][[algo]][!is.na(m[[i]][[algo]])]<0.1)
    # sum((m[[i]][[algo]] < alpha))
  })))
}
```

```

}

getSensitivityAlgo <- function(m, alpha, alphaOut) {
  t(sapply(1:nreps, function(i) sapply(namesAlgosALL, function(algo) {
    sigHeldout <- resHeldout[[m]][[i]][[algo]] <= alpha
    mean((resTes[[m]][[i]][[algo]] <= alpha)[sigHeldout])
  })))
}

getPrecisionAlgo <- function(m, alpha, alphaOut) {
  t(sapply(1:nreps, function(i) sapply(namesAlgosALL, function(algo) {
    sigTest <- resTes[[m]][[i]][[algo]] <= alpha #Evaluation set
    if (sum(sigTest) == 0) return(0)
    mean((resHeldout[[m]][[i]][[algo]] <= alpha)[sigTest]) #TP rate
  })))
}

getSensitivityAlgoGold <- function(m, alpha, alphaOut, gold) {
  t(sapply(1:nreps, function(i) sapply(namesAlgosALL, function(algo) {
    sigHeldout <- resHeldoutALL[[m]][[i]][[gold]] < alphaOut
    mean((resTestALL[[m]][[i]][[algo]] < alpha)[!is.na(sigHeldout)])
  })))
}

getPrecisionAlgoGold <- function(m, alpha, alphaOut, gold) {
  t(sapply(1:nreps, function(i) sapply(namesAlgosALL, function(algo) {
    sigTest <- resTestALL[[m]][[i]][[algo]] < alpha
    if (sum(sigTest) == 0) return(0)
    mean((resHeldoutALL[[m]][[i]][[gold]] < alphaOut)[!is.na(sigTest)])
  })))
}

```

The following function helps to rename algorithms.

```

renameAtoB <- function(f,a,b) {
  levels(f)[levels(f) == a] <- b
  f
}

```

```

namesAlgos <- make.names(names(resTes[[1]]))
names(namesAlgos) <- namesAlgos

```

```

library(RColorBrewer)
# display.brewer.all()

cols <- c(
  # DEseq
  brewer.pal(n = 9, "YlOrRd")[c(3,4,5,6)],
  # Edger
  brewer.pal(n = 9, name = "GnBu")[c(5,6,7)],
  # limma
  brewer.pal(n = 9, "RdPu")[c(5)],
  # circMeta
  brewer.pal(n = 9, "OrRd")[c(8)],

```

```

# GLMM
brewer.pal(n = 9, "BuPu")[c(5)]

)

methods2 <- c("DESeq2",
              "DESeq2-ZeroInflated",
              "DESeq2-GamPoi",
              "DESeq2-ZINB Wave",
              "edgeR",
              "edgeR-robust",
              "edgeR-ZINB Wave",
              "voom",
              "circMeta",
              "GLMM")

names(cols) <- methods2

```

2 Counting number of calls

Here we produce boxplots of the number of calls based on adjusted p -value for each algorithm in the evaluation set and verification set for each random replicate.

```
nreps <- length(resHeldout)
nalgo <- length(namesAlgos)
# nmethods <- length(names(resHeldout))

heldMat <- getCalls(m = resHeldout, alpha = .1)
testMat <- getCalls(m = resTes, alpha = .1)

d <- data.frame(heldoutCalls=reshape2::melt(heldMat)$value,
               testCalls=reshape2::melt(testMat)$value,
               algorithm=factor(rep(namesAlgos,each=nrow(heldMat)),
                                levels=namesAlgos))
               #detection.method=rep(factor(rep(names(resHeldout), each=nrow(heldMat))), nalgo))

d$algorithm <- renameAtoB(d$algorithm, "edgeR.robust", "edgeR-robust")
d$algorithm <- renameAtoB(d$algorithm, "edgeR.ZINBWave", "edgeR-ZINB Wave")
d$algorithm <- renameAtoB(d$algorithm, "DESeq2.ZINBWave", "DESeq2-ZINB Wave")
d$algorithm <- renameAtoB(d$algorithm, "DESeq2.ZI", "DESeq2-ZeroInflated")
d$algorithm <- renameAtoB(d$algorithm, "DESeq2.glmGamPoi", "DESeq2-GamPoi")
d$algorithm <- renameAtoB(d$algorithm, "GLMM_NB", "GLMM")

d$algorithm <- as.character(d$algorithm)

d <- d %>%
  mutate_if(sapply(d, is.character), as.factor)
d$heldoutCalls <- as.numeric(d$heldoutCalls)
d$testCalls <- as.numeric(d$testCalls)
```

```
p <- ggplot(d, aes(x=reorder(algorithm,testCalls,median),y=testCalls,color=algorithm))
# png("/blackhole/alessia/CircModel/robustness/Figure/DEcalls_evaluation_ALZ.png", res = 200, units = "cm")
p + geom_boxplot(outlier.colour=rgb(0,0,0,0)) + theme_bw() +
  geom_point(position = position_jitter(w = 0.1, h = 0), color="grey50", size=1) +
  # facet_grid(.~detection.method) +
  scale_color_manual(values = cols, guide="none") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 12),
        axis.text.y = element_text(size = 9),
        axis.text.y.left = element_text(size = 12),
        strip.text.x = element_text(size = 9)) + xlab("") +
  ylab("number of DE calls in Evaluation set")
```

```
# dev.off()
```

```
p <- ggplot(d, aes(x=reorder(algorithm,heldoutCalls,median),y=heldoutCalls,color=algorithm))
# png("/blackhole/alessia/CircModel/power/Figure/DEcalls_verification_ALZ.png", res = 200, units = "cm", w
p + geom_boxplot(outlier.colour=rgb(0,0,0,0)) + theme_bw() +
  geom_point(position = position_jitter(w = 0.1, h = 0), color="grey50", size=1) +
  # facet_grid(.~detection.method) +
  scale_color_manual(values = cols, guide="none") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 12),
        axis.text.y = element_text(size = 9),
        axis.text.y.left = element_text(size = 12),
```

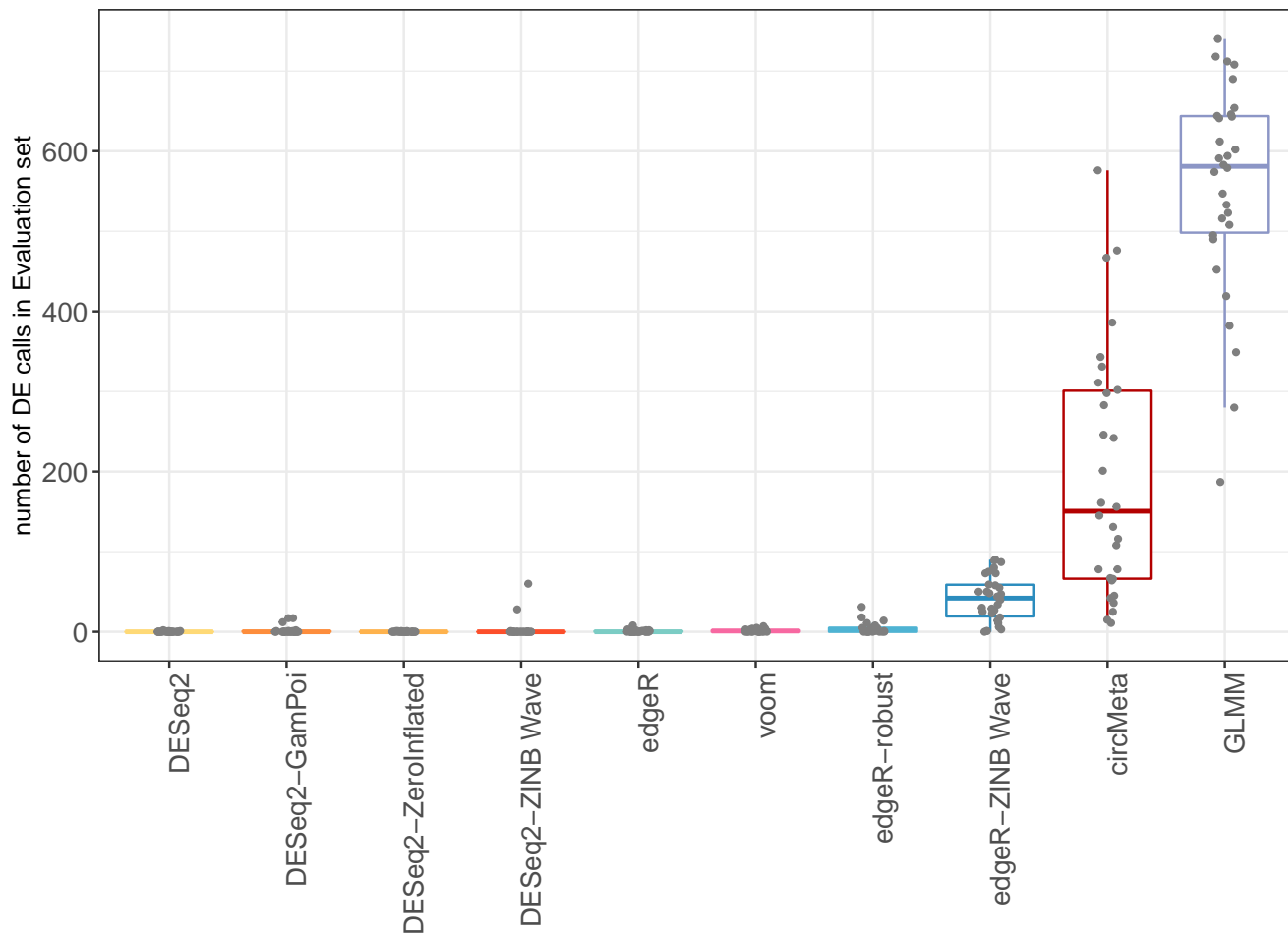


Figure 1: Evaluation set calls (adjusted p -value $< .1$)

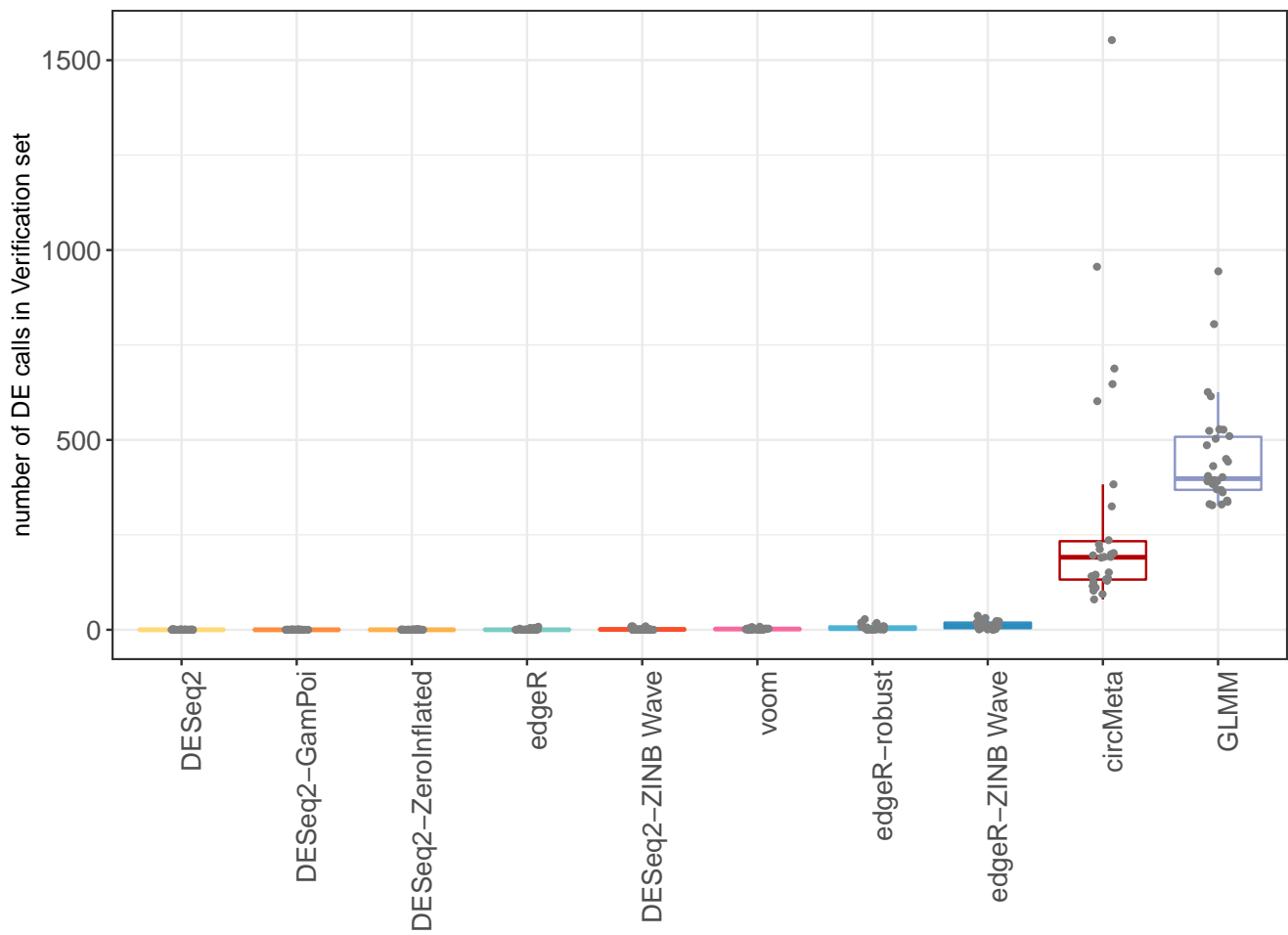


Figure 2: Verification set calls (adjusted p -value $< .1$)

```
strip.text.x = element_text(size = 9)) + xlab("") +
ylab("number of DE calls in Verification set")
```

```
# dev.off()
```

3 Sensitivity and precision plots

We construct a data frame containing the sensitivity and precision estimates for every algorithm in the evaluation set and the verification set.

```
nreps = 30
gold = "GLMM_NB"
alpha = .1
alphaOut = .1
resHeldout = lapply(resHeldout, function(x){z=x[complete.cases(x),]
return(z)})
resTes = lapply(resTes, function(x){z=x[complete.cases(x),]
return(z)})

res = t(sapply(1:nreps, function(i) sapply(namesAlgos, function(algo) {
  sigHeldout <- resHeldout[[i]][[gold]] < alphaOut
  mean((resTes[[i]][[algo]] < alpha)[sigHeldout])
})))

sensMat = data.frame(res, heldout=rep("GLMM",nrow(res)))

sensMelt <- suppressWarnings(melt(sensMat, id=c("heldout")))
names(sensMelt) <- c("verification", "evaluation","sensitivity")

res = t(sapply(1:nreps, function(i) sapply(namesAlgos, function(algo) {
  sigTest <- resTes[[i]][[algo]] < alpha
  if (sum(sigTest) == 0) return(0)
  mean((resHeldout[[i]][[gold]] < alphaOut)[sigTest])
})))
precMat = data.frame(res, heldout=rep("GLMM", nrow(res)))

precMelt <- suppressWarnings(melt(precMat, id=c("heldout")))
names(precMelt) <- c("verification", "evaluation","precision")

d <- data.frame(sensMelt, precision=precMelt$precision)
d$evaluation <- factor(d$evaluation)
d$verification <- factor(d$verification)

d$evaluation <- renameAtoB(d$evaluation, "edgeR.robust", "edgeR-robust")
d$verification <- renameAtoB(d$verification, "edgeR.robust", "edgeR-robust")
d$verification <- renameAtoB(d$verification, "edgeR.ZINBWave", "edgeR-ZINB Wave")
d$evaluation <- renameAtoB(d$evaluation, "edgeR.ZINBWave", "edgeR-ZINB Wave")
d$evaluation <- renameAtoB(d$evaluation, "DESeq2.ZINBWave", "DESeq2-ZINB Wave")
d$verification <- renameAtoB(d$verification, "DESeq2.ZINBWave", "DESeq2-ZINB Wave")
d$evaluation <- renameAtoB(d$evaluation, "DESeq2.glmGamPoi", "DESeq2-GamPoi")
d$verification <- renameAtoB(d$verification, "DESeq2.glmGamPoi", "DESeq2-GamPoi")
d$evaluation <- renameAtoB(d$evaluation, "DESeq2.ZI", "DESeq2-ZeroInflated")
d$verification <- renameAtoB(d$verification, "DESeq2.ZI", "DESeq2-ZeroInflated")
d$evaluation <- renameAtoB(d$evaluation, "GLMM_NB", "GLMM")
d$verification <- renameAtoB(d$verification, "GLMM_NB", "GLMM")
d <- d %>%
  mutate_if(sapply(d, is.character), as.factor)
# d$verification <- renameAtoB(d$verification, "GLMM", "GLMM-ZINB")
# d$evaluation <- renameAtoB(d$evaluation, "GLMM", "GLMM-ZINB")
```

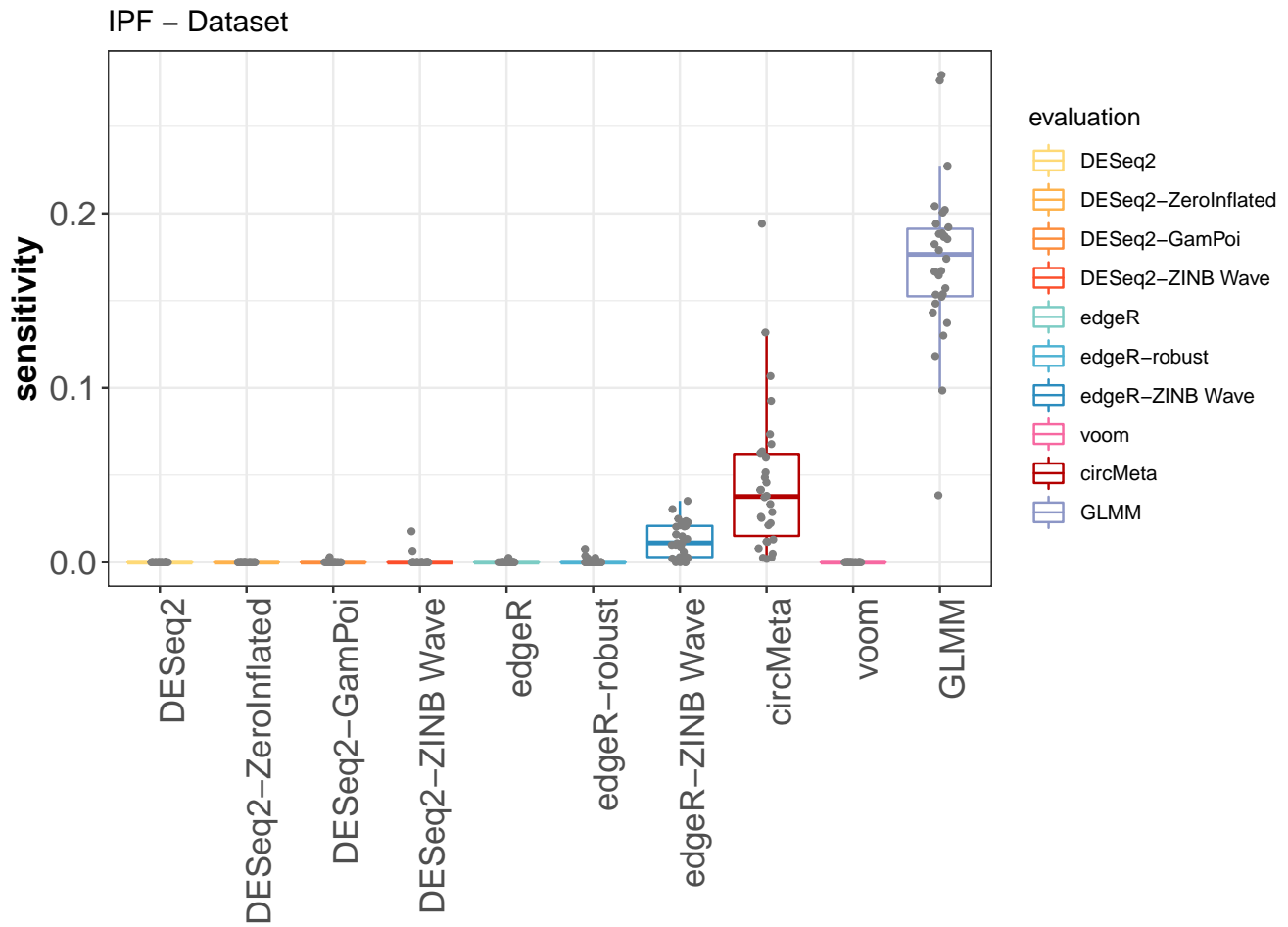


Figure 3: Sensitivity, where each algorithm's calls (adjusted p -value $< .1$) in the evaluation set (color boxes) is compared against itself (adjusted p -value $< .1$) in the verification set (grey labels).

```
# d.glm <- d.glm[complete.cases(d.glm),]
# png("/blackhole/alessia/CircModel/power/Figure/Power_sensitivity_ALZ.png", res = 200, units = "cm", width = 10, height = 10)
p <- ggplot(d, aes(x=evaluation,y=sensitivity,color=evaluation))
p + geom_boxplot(outlier.colour=rgb(0,0,0,0)) + theme_bw() +
  # facet_wrap(~ detection.method) +
  geom_point(position = position_jitter(w = 0.1, h = 0), color="grey50", size=1) +
  scale_color_manual(values=cols) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 15),
        axis.text.y = element_text(size = 15),
        axis.title=element_text(size=15,face="bold"),
        # axis.text.y.left = element_text(size = 11),
        strip.text.x = element_text(size = 13)) +
  ggtitle("IPF - Dataset") +
  xlab("")

# dev.off()
```

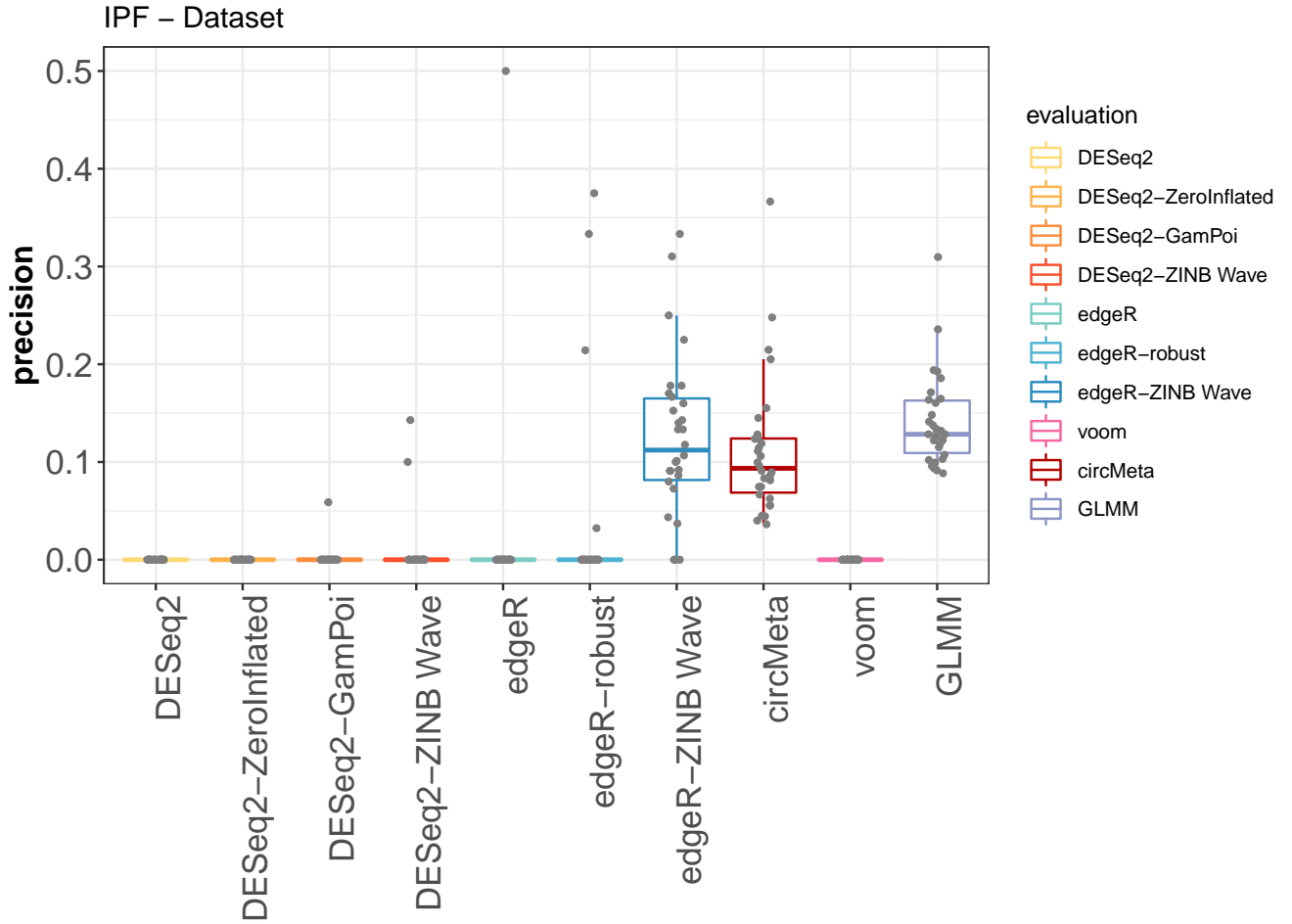
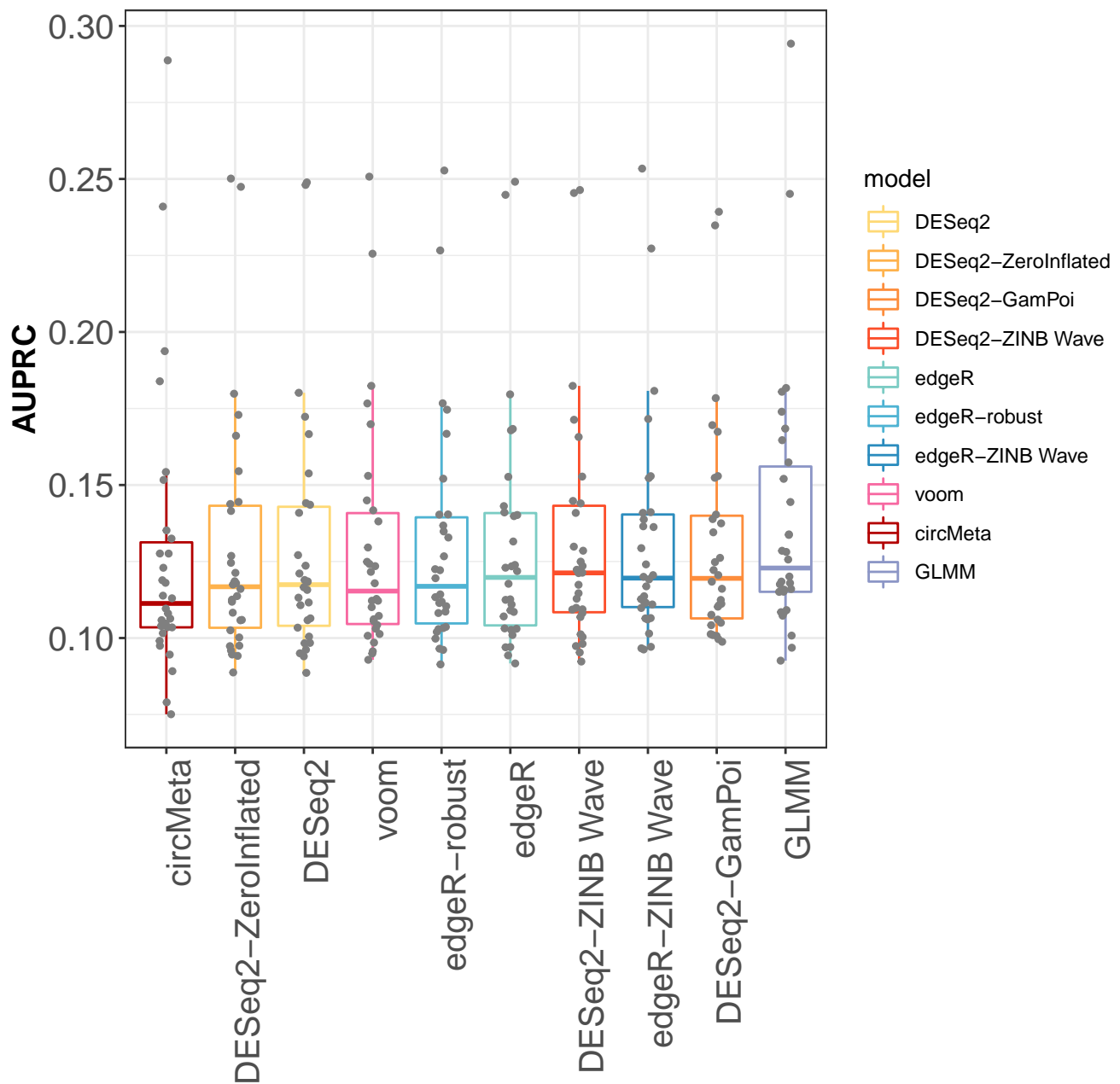



Figure 4: Precision, where each algorithm's calls (adjusted p -value $< .1$) in the evaluation set (color boxes) is compared against itselfs (adjusted p -value $< .1$) in the verification set (grey labels).

```
# png("/blackhole/alessia/CircModel/power/Figure/Power_precision_ALZ.png", res = 200, units = "cm", width
p <- ggplot(d, aes(x=evaluation,y=precision,color=evaluation))
p + geom_boxplot(outlier.colour=rgb(0,0,0,0)) + theme_bw() +
  # facet_wrap(~ detection.method) +
  scale_color_manual(values = cols) +
  geom_point(position = position_jitter(w = 0.1, h = 0), color="grey50", size=1) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 15),
        axis.text.y = element_text(size = 15),
        axis.title=element_text(size=14,face="bold"),
        # axis.text.y.left = element_text(size = 11),
        strip.text.x = element_text(size = 13)) +
  ggtitle("IPF - Dataset") +
  xlab("")

# dev.off()
```

4 Area Under Precision Recall Curve (AUPRC)



5 Clustering of calls

```
alpha <- .1
library("abind")
library("ggplot2")
library("ggdendro")
library("cowplot")

##
## Attaching package: 'cowplot'
```

```

## The following object is masked from 'package:reshape':
##
##      stamp

library("gridExtra")

##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##      combine

library("grid")
methods <- c("DESeq2",
             "DESeq2.ZI",
             "DESeq2.glmGamPoi",
             "DESeq2.ZINBWave",
             "edgeR",
             "edgeR.robust",
             "edgeR.ZINBWave",
             "circMeta",
             "voom",
             "GLMM_NB")

names(cols) <- methods
# first with evaluation sets
nreps = 30
j0 <- lapply(1:nreps, function(i) {
  # i=1
  #print(i)
  dt <- resTes[[i]]
  colnames(dt) <- c(colnames(dt)[c(1:7)], "voom", "circMeta",
                    "GLMM")
  m <- as.matrix(dist(t(dt) <= alpha), method="binary")
  return(m)
})

j <- abind(j0,along=3)
average.Jaccard <- apply(j,c(1,2),mean)
average.Jaccard <- as.dist(average.Jaccard)
hcTest <- hclust(average.Jaccard)

# again with verification sets
j0 <- lapply(1:nreps, function(i) {
  dt <- resHeldout[[i]]
  colnames(dt) <- c(colnames(dt)[c(1:7)], "voom", "circMeta",
                    "GLMM")
  m <- as.matrix(dist(t(dt) <= alpha), method="binary")
  return(m)
})
j <- abind(j0,along=3)
average.Jaccard <- apply(j,c(1,2),mean)
average.Jaccard <- as.dist(average.Jaccard)
hcHeldout <- hclust(average.Jaccard)

```

```

g_horizontal_dendrogram_Heldout <- ggplot() +
  geom_segment(data=dendro_data(hcHeldout)$segments, aes(x=x, y=y, xend=xend, yend=yend)) +
  geom_label(data=dendro_data(hcHeldout)$labels, aes(x=x, y=y, label=label,
                                                    hjust=-0.05,color=label), nudge_y = 0) +

  coord_flip() + scale_y_reverse(expand = c(0,0,0,0)) + scale_x_reverse() +
  scale_color_manual(values = cols) +
  theme(axis.line.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.line.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.title.y=element_blank(),
        axis.text.x=element_blank(),
        axis.title.x=element_blank(),
        panel.background=element_rect(fill="white"),
        panel.grid=element_blank(),
        legend.position = "none",
        panel.spacing = unit(0, "lines"),
        plot.margin = unit(c(0.5,0.5,0.5, 0.5), "cm")) +
  expand_limits(y = -round(max(segment(dendro_data(hcHeldout))$y) * 0.5, 1))

get_legend_dendo <- get_legend(ggplot() +
  geom_segment(data=dendro_data(hcHeldout)$segments, aes(x=x, y=y, xend=xend, yend=yend)) +
  geom_point(data=dendro_data(hcHeldout)$labels, aes(x=x, y=y,color=factor(label, levels = names(cols),
  scale_y_continuous() +
  #scale_y_reverse(expand=c(2,1)) + scale_x_reverse(expand=c(2,1)) +
  scale_color_manual(values = cols) +
  theme(axis.line.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.line.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.title.y=element_blank(),
        axis.text.x=element_blank(),
        axis.title.x=element_blank(),
        panel.background=element_rect(fill="white"),
        panel.grid=element_blank(),
        legend.position = "bottom",
        panel.spacing = unit(0, "lines"),
        plot.margin = unit(c(0.5,0.5,0.5, 0.5), "cm")) +
  guides(color = guide_legend(title = "Methods:",title.position = "left",nrow = 2)))

# now add the title
title <- ggdraw() +
  draw_label(
    "Clustering of calls in Verification set",
    fontface = 'bold',
    x = 0,
    hjust = 0, size = 12
  ) +
  theme(
    # add margin on the left of the drawing canvas,
    # so title is aligned with left edge of first plot
    plot.margin = margin(0, 0, 0, 7)
  )

```

```

g_horizontal_dendrogram_Test <- ggplot() +
  geom_segment(data=dendro_data(hcTest)$segments, aes(x=x, y=y, xend=xend, yend=yend)) +
  geom_label(data=dendro_data(hcTest)$labels, aes(x=x, y=y, label=label, hjust=-0.05,color=label), nudge
  coord_flip() + scale_y_reverse(expand = c(0,0,0,0)) + scale_x_reverse() +
  scale_color_manual(values = cols) +
  theme(axis.line.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.line.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.title.y=element_blank(),
        axis.text.x=element_blank(),
        axis.title.x=element_blank(),
        panel.background=element_rect(fill="white"),
        panel.grid=element_blank(),
        legend.position = "none",
        panel.spacing = unit(0, "lines"),
        plot.margin = unit(c(0.5,0.5,0.5, 0.5), "cm")) +
  expand_limits(y = -round(max(segment(dendro_data(hcTest))$y) * 0.5, 1))

# now add the title
titleTest <- ggdraw() +
  draw_label(
    "Clustering of calls in Evaluation set",
    fontface = 'bold',
    x = 0,
    hjust = 0, size = 12
  ) +
  theme(
    # add margin on the left of the drawing canvas,
    # so title is aligned with left edge of first plot
    plot.margin = margin(0, 0, 0, 7)
  )

Heldplot <- plot_grid(title, g_horizontal_dendrogram_Heldout,
  # plot_grid(plotlist = g_horizontal_dendrogram_Heldout,
  #           ncol = 2, #align = 'vh',
  #           hjust = c(-0.1, 0.8, -0.1, 0.8, -0.1, 0.8, -0.1),
  #           vjust = -1.35,
  #           labels = names(heldMat),
  #           label_size = 9),
  ncol = 1, rel_heights = c(0.1, 1))
Testplot <- plot_grid(titleTest, g_horizontal_dendrogram_Test,
  # plot_grid(plotlist = g_horizontal_dendrogram_Test,
  #           ncol = 2, #align = 'vh',
  #           hjust = c(-0.1, 0.8, -0.1, 0.8, -0.1, 0.8, -0.1),
  #           vjust = -1.35,
  #           labels = names(testMat),
  #           label_size = 9
  #           ),
  ncol = 1, rel_heights = c(0.1, 1))
# png("/blackhole/alessia/CircModel/robustness/ALZ_Clusteringofcalls.png", res = 200, units = "cm", width
grid.arrange(Heldplot, Testplot, get_legend_dendo, ncol=2, nrow=2, widths=c(2.3, 2.3), heights=c(1, 0.1))

```

Clustering of calls in Verification set

Clustering of calls in Evaluation set

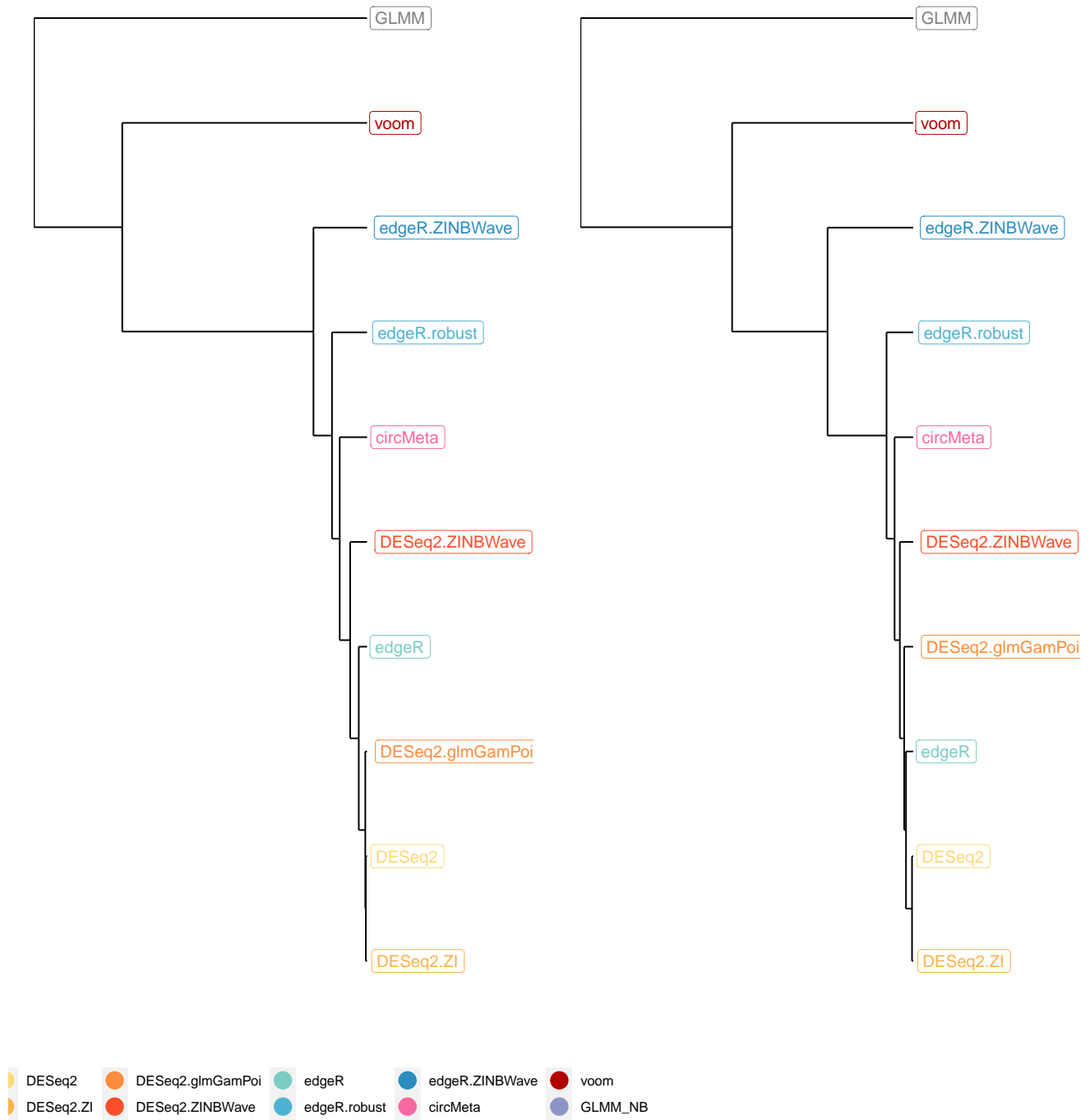


Figure 5: Clustering of calls (adjusted p -value $< .1$) with distances based on the Jaccard index

```
# dev.off()
```

6 Session information

- R version 4.1.1 (2021-08-10), x86_64-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=en_US.UTF-8, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Running under: Ubuntu 20.04.3 LTS
- Matrix products: default
- BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
- LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, utils
- Other packages: abind 1.4-5, cowplot 1.1.1, data.table 1.14.0, dplyr 1.0.7, ggdendro 0.1.22, ggplot2 3.3.5, gridExtra 2.3, knitr 1.33, PRROC 1.3.1, RColorBrewer 1.1-2, reshape 0.8.8
- Loaded via a namespace (and not attached): assertthat 0.2.1, colorspace 2.0-2, compiler 4.1.1, crayon 1.4.1, DBI 1.1.1, digest 0.6.27, ellipsis 0.3.2, evaluate 0.14, fansi 0.5.0, farver 2.1.0, generics 0.1.0, glue 1.4.2, gtable 0.3.0, highr 0.9, labeling 0.4.2, lifecycle 1.0.0, magrittr 2.0.1, MASS 7.3-54, munsell 0.5.0, pillar 1.6.2, pkgconfig 2.0.3, plyr 1.8.6, pROC 1.17.0.1, purrr 0.3.4, R6 2.5.0, Rcpp 1.0.7, reshape2 1.4.4, rlang 0.4.11, scales 1.1.1, stringi 1.7.3, stringr 1.4.0, tibble 3.1.3, tidyselect 1.1.1, tools 4.1.1, utf8 1.2.2, vctrs 0.3.8, withr 2.4.2, xfun 0.25